MULTI-VIEW GRAPH CLUSTERING WITH PARTIAL INFORMATION DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-view graph clustering aims to partition graph elements, including graphs and nodes, by exploring diverse views with distinct semantic topology patterns. Existing methods obtain discriminative representations mainly by integrating multi-view features or modeling consistency across views. However, these strategies assume cross-view information is entirely shared, which ignores other possible types of interaction, degrading the robustness and applicability of clustering. To this end, we propose a multi-view graph clustering framework based on partial information decomposition (PIDGC) that enables communication between multiple graph views in a universal space. Instead of enforcing restrictions within or across views, we achieve alignment of multiple graph views by optimizing the mutual information between original views and corresponding augmented counterparts, exploring redundant interactions between different views. We provide a theoretical illustration to demonstrate that mutual information can naturally explore diverse cross-view information via partial information decomposition, improving the discrimination of representations and clustering performance. Extensive experiments on multifarious datasets demonstrate the effectiveness and advancement of PIDGC compared with state-of-the-art baselines.

1 INTRODUCTION

Graph clustering (Han et al., 2026) aims to partition elements, such as nodes and graphs, into different groups, which integrates structural topology encoding with information propagation to obtain high-quality representations, serving as a fundamental task for understanding and improving web-based systems. However, traditional graph clustering focuses on a single view, which ignores rich semantic information in complex topologies. Recently, multi-view graph clustering (MVGC) (Liang et al., 2025; Wang et al., 2026b) has become ubiquitous and has drawn extensive attention in various applications, including bioinformatics (Xie et al., 2026) and multi-modal analytics (Tang et al., 2024), because different views may signify distinct semantic and relationship patterns. Beyond the traditional relational adjacency topology, existing methods focus on exploring diverse latent topology views to enhance the semantic expression of graphs via data augmentation (Zhao et al., 2025) and similarity evaluation (Ren et al., 2025), supplementing richer semantic information for discriminative representations. Besides, some studies explore the interactions between different views via topology fusion (Wang et al., 2026b) and variational sampling (Chen et al., 2025) to further improve the consistency information between different views.

However, the effectiveness of existing multi-view clustering models often focuses on a fairly strong assumption: cross-view information related to the clustering objective is entirely shared. The intuition seems supportive at first glance because the representations from different views represent the same samples, which should exhibit high consistency. Essentially, entirely shared information indicates exploiting multi-view consistency to aid representation learning on a single view. Therefore, existing paradigms predefine a single and vanilla relationship between different views, immensely restricting multiple possible alternatives of the relationship between different views. However, clustering-discriminative cues in diverse views do not present a single relationship. In many cases, they appear only in some views, or become evident only when information across views is combined. Once the strong assumption no longer holds, the robustness and applicability of these methods rapidly degrade.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To further illustrate and interpret the interaction between different views, we explore multiple relationships for cross-view interactions. Inspired by Partial Information Decomposition (PID) (Bertschinger et al., 2014), we introduce three types of information: **(1) Redundancy (R)** occurs when multi-view clustering can be performed using any single view because the views contain redundant information; **(2) Uniqueness (U)** emerges when one view contains all imperative and critical information for the task; **(3) Synergy (S)** occurs when multiple views are needed simultaneously to contribute view-specific clustering. However, extracting these interactions is overly strenuous since they have no distinct boundaries in complex real-life applications and exhibit strong dependence on downstream tasks. Therefore, to obtain view-specific interactions for multi-view clustering tasks, models need to simultaneously capture R, U, and S to learn more discriminative and interpretable representations.

To achieve this goal, we propose a multi-view graph clustering framework based on partial information decomposition (PIDGC) that enhances multiple interactions across different views. Similar to traditional multi-view graph clustering, we also integrate the view-specific representations to generate the final embeddings. However, unlike traditional methods that simply align multi-view representations under the strong assumption of redundancy, the novelty of PIDGC is that it explores synergy and redundancy interactions by calculating the mutual information between augmented views, and captures uniqueness interactions by optimizing the mutual information between multiple views and their augmented representations. The effectiveness and rationality lie in the theoretical analysis based on PID, which reveals that the mutual information between specific representations can capture multiple interactions across views. Our contributions are as follows:

- **A new problem**. We make the first attempt to explore the multiple possible relationships between different graph views for graph clustering, alleviating the strong single-relationship assumption across different views.
- **A novel framework**. We propose PIDGC that utilizes contrastive learning with augmentations to capture multiple interactions between different views for modeling alignment.
- **Advanced performance**. Extensive experiments on diverse public datasets demonstrate the superior generalization of PIDGC and cross-granularity detection ability compared to state-of-the-art baselines.

2 Related Work

Graph clustering aims to achieve a partition of specific elements, which is widely applied in various fields. Existing graph clustering mainly includes node-level clustering and graph-level clustering. Node-level clustering focuses on achieving node partitions via discriminative node representations learned by graph encoders. For example, SGSF-GC (Zhang et al., 2026c) introduces different graph filters to explore view-specific semantics. Graph-level clustering focuses on discovering meaningful groups of graphs by characterizing their underlying structural and semantic similarities. Representative methods, including MCCL (Sun et al., 2026b), SL-GSSL (Amar et al., 2026), and GrePool (Liu et al., 2025), learn graph representations by encoding local node information and aggregating it into graph-level embeddings. Although node-level and graph-level clustering have different downstream tasks, they both aim to achieve high-quality element partition via graph encoding technologies. However, they usually focus on a single graph view, missing important complementary views.

Recently, many studies introduce multi-view graphs to capture more redundant semantic information to improve the representation learning ability. CVMP (Quan et al., 2025) dynamically constructs node-specific cross-view edges based on node features and structural information. Representation learning and hyperspherical clustering are further unified in an end-to-end framework. RK-MVC (Liang et al., 2025) compensates for the omitted sample-global correlations from the structural knowledge information, generating sample-global correlations in both coarse-grained and fine-grained manners and fusing them with basic consensus features as the target consensus feature. TOTF (Li et al., 2025) trains missing-aware encoders first to capture view-specific information, and then freezes their parameters to discover common information available for clustering. To handle the unreliability of individual views, variational inference (Chen et al., 2025) and uncertainty estimation (Peng et al., 2026) have been introduced into MVGC, which weigh views or samples by their inferred credibility rather than fixed weights. These methods usually rely on contrastive learning to

model alignment between different views. However, they usually are based on a strong assumption that cross-view interaction is only one type, ignoring other possibilities for interactions. In this study, we explore diverse interactions between different views to improve the alignment.

3 PRELIMINARIES

Notations and Statement. Denote a graph dataset $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \cdots \mathcal{G}_N\}$, where $N = |\mathcal{G}|$ is the number of graph samples. Denote $\mathcal{G}_i = \{\mathcal{V}_i, X_i, A_i\}$ as i-th graph. $\mathcal{V}_i$ is the node set of $\mathcal{G}_i$ and $|\mathcal{V}_i| = B_i$. $X_i \in \mathbb{R}^{B_i \times F}$ and $A_i \in \{0,1\}^{B_i \times B_i}$, respectively, represent the node feature matrix and adjacency matrix, where B_i and F denote the number of nodes in $\mathcal{G}_i$ and the original feature dimension. For graph-level clustering, the aim is to partition the graph set $\mathcal{G}$ into m independent clusters $\mathcal{C} = \{C_1, \cdots, C_m\}$ and $\sum_i^m |C_i| = N$. For node-level clustering, the aim is to partition the node set $\mathcal{V}_i$ or into m independent clusters $\mathcal{C} = \{C_1, \cdots, C_m\}$ and $\sum_i^m |C_i| = B_i$.

Partial Information Decomposition (PID). Denote a downstream task as Y, PID illustrates that the mutual information $I(X_1, X_2; Y)$ can be divided into three components:

- **Uniqueness.** This item indicates that the task Y can be performed with one single granularity. U_1 (U_2) refers to the task-specific information in $X_1(X_2)$.
- **Redundancy.** This item indicates the shared or redundant information R about task Y in both X_1 and AX_2.
- **Synergy.** This item indicates that X_1 and X_2 hold different task-related information.

Therefore, the following equations can be derived according to the chain rule of mutual information:

$$I(X_1, X_2; Y) = R+S+U_1+U_2, I(X_1; Y) = R+U_1, I(X_2; Y) = R+U_2, I(X_1; X_2; Y) = R-S, \tag{1}$$

Assumption 1 *$\exists \sigma > 0$ such that $I(Y; X_1|X_2) < \sigma$ and $I(Y; X_2|X_1) < \sigma$.*

Assumption 1 indicates that most task-specific information is shared between graph views and that the unshared information is not larger than σ. The intuition of Assumption 1 is that representations of different views can be obtained using the same features. Therefore, they have the same origin and can provide adequate guidance to other views.

Lemma 1 *Given the redundancy assumption, cross-view contrastive learning can only learn the redundant information R.*

We illustrate the situation where multiple sources of information from different views interpose. A straightforward approach would be to introduce dedicated mechanisms to integrate distinct and shared patterns into the learning framework, for instance, by designing view-specific augmentations or conditional transformations. However, such strategies often rely on strong assumptions about the interactions between multiple views, which may not hold universally. In practice, these assumptions can lead to brittle designs that exacerbate error accumulation and are sensitive to the choice of augmentation strategies, especially when cross-view relationships are complex or dynamically changing. Therefore, while these approaches offer an intuitive solution, they risk oversimplifying the problem and fail to capture the nuanced interplay between views, ultimately compromising the robustness and discrimination of the learned encoding.

Assumption 2 *We denote an augmentation strategy set $\mathcal{T}^*$, for any $t \in \mathcal{T}^*$, we have $X^{'} = t(X)$, then we obtain $I(X, X^{'}) = I(X, Y)$.*

4 CONTRASTIVE MULTI-VIEW GRAPH CLUSTERING

The objective is to obtain multi-view representations that are discriminative enough for graph clustering tasks. Contrastive learning has shown powerful modeling capacity and exhibits promising prospects in learning high-quality representations. However, according to Section 3, they struggle

with the strong assumption of shared cross-view information, failing to explore diverse interplay between multiple views. In this section, we propose a solution based on multi-view graph contrastive learning. The main components are twofold:

- A contrastive learning framework for multi-view graph clustering that utilizes augmented representations to achieve alignment and adaptively integrate view-specific features to generate the integrated representations for clustering.
- A contrastive objective that explores redundancy, uniqueness, and synergy interactions between representations from different views.

4.1 Discriminating Multi-View Representations

Existing studies focus on node-edge granularity unification and node-graph granularity unification. Without loss of generality, we define two representations A_1, A_2 under different granularities. To derive latent and discriminative representations H that extract uniqueness, redundancy, and synergy from different granularities, we develop an encoder f_θ with learnable parameters θ such that $H_\theta = f_\theta(X, A_i)$ for $i = 1, 2$. We define $X^{'} = t(X)$ where $t \in \mathcal{T}$ is traditional graph augmentation strategies.

According to the data flow $X \to X' \to H_\theta$ and $H_\theta^{'} \to X \to H_\theta^{'}$, we can obtain:

$$I(H_\theta, H_\theta^{'}) \leq I(X, H_\theta^{'}) \leq I(X, X^{'}) \tag{2}$$

Therefore, we can derive the following conclusion.

Lemma 2 *Through optimizing the encoder f_θ based on $I(H_\theta, H_\theta^{'})$, we obtain:*

$$I(H_{\theta^*}, H_{\theta^*}^{'}) = I(X, X^{'}) \tag{3}$$

where θ^* is the optimal parameter.

Theorem 1 *Let f_{θ^*} denote the optimal solution for $I(H_\theta, H_\theta^{'})$ optimization. Then, we can obtain $I(H_{\theta^*}, Y) = I(X, Y)$. For a specific view A_i, consider an augmentation strategy $t \in' \mathcal{T}$ such that $H_i^{'} = f_{\theta^*}(t(X), A_i)$, we have:*

$$I(H_i^{'}, Y) = R + U_i. \tag{4}$$

Theorem 1 indicates that the optimal embeddings H_i hold all view-specific information. All proofs are in Section A. For practical implementation, we analyze the key components for contrastive learning to achieve the extraction of multi-view interactions.

- Given Lemma 2 and Assumption 2, we can obtain $I(X, Y) = I(H_\theta, H_\theta^{'})$. Therefore, we can directly utilize $I(H_\theta, H_\theta^{'})$ to learn $I(X, Y) = U_1 + U_2 + R + S$.
- Based on Theorem 1, for any view i, we can leverage $I(H_\theta, H_\theta^{'})$ to model $R + U_i$.

4.2 Detailed Architecture

The framework is shown in Fig. 1. To extract multi-view interactive information, the model is comprised of three components: view generator, encoders, and concentrator.

View Generator. This module generates different views beyond the original relational topology. Specifically, we generate a supplementary topology using cosine similarity, and employ a top-k function with a hyperparameter K to select neighbors for each node and generate the complementary graph view.

Encoders. We utilize GNNs as basic encoders to obtain latent representations, and each view is encoded individually by a view-specific encoder.

Concentrator. It is used to concatenate the latent representations from different views to generate integrated observations for the downstream clustering task.

4.3 Training and Inference

In this section, we first introduce the training of multi-view PID individually for node- and graph-level representations. Then, we introduce the clustering loss.

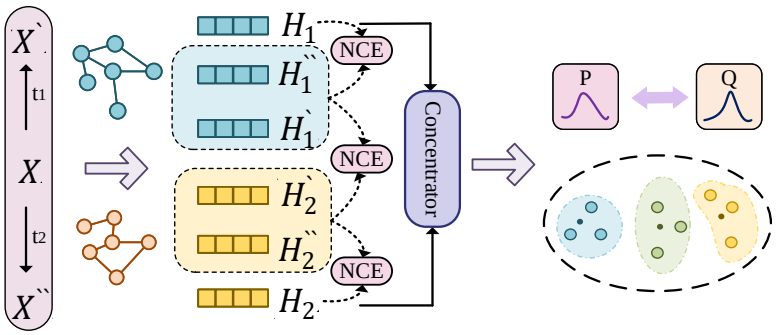


Figure 1: The training process of MVGCPI.

Multi-View Node-Level Encoding. Given a graph $G = \{\mathcal{V}, X, A_1\}$, we utilize two augmentation strategies t_1, t_2 to obtain the masked features $X' = t_1(X)$ and $X'' = t_2(X)$ and then generate two different views $H_1^{'} = f_{\theta_1}(X', A_1), H_2^{'} = f_{\theta_2}(X', A_2), H_1^{''} = f_{\theta_1}(X'', A_1), H_2^{''} = f_{\theta_2}(X'', A_2)$. Also, we consider the latent encoding without augmentation $H_1 = f_{\theta_1}(X, A_1), H_2 = f_{\theta_2}(X, A_2)$. After obtaining $H_i^{'}, H_i^{''}, H_i (i = 1, 2)$, we calculate the mutual information to optimize the model parameters via backpropagation. Specifically, $I(H_i^{'}, H_i^{''})$ aims to optimize $\sum_i U_i + R + S$, while $I(H_i, H_i^{'})$ $I(H_i, H_i^{''})$ aims to optimize $R + U_i$.

We employ a learnable parameter $\alpha \in \mathbb{R}^2$ to allocate adaptive weights for each view and obtain the integrated representations $Z \in \mathbb{R}^{B \times d}$:

$$Z = \frac{\sum_{i=1}^{2} \alpha_i H_i}{\sum_{i=1}^{2} \alpha_i}. \tag{5}$$

Multi-View Graph-Level Encoding. Denote $\mathcal{H} = \{H_i^{'}, H_i^{''}, H_i\}(i = 1, 2)$. We leverage average pooling to generate global representations $h_i^g = \text{Mean}(h_i)(h_i \in \mathcal{H})$ for each graph, and then we obtain $h_i^{'(g)}, h_i^{''(g)}, h_i^g$. Similar to node-level multi-view PID, we calculate $I(h_i^{'(g)}, h_i^{''(g)})$, $I(h_i^g, h_i^{'(g)})$, and $I(h_i^g, h_i^{''(g)})$ to explore the interactions between different views for graph-level representations.

For graph-level clustering, denote $H_i^g \in \mathbb{R}^{N \times d}$ as the i-th view graph-level representations for the graph dataset $\mathcal{G}$. Similarly, we leverage $H_i^g \in \mathbb{R}^{N \times d}$ to generate integrated representations $Z \in \mathbb{R}^{N \times d}$.

PID Loss. We employ InfoNCE (Han et al., 2026) to approximate mutual information since it has powerful modeling ability and simple calculation:

$$I_{NCE}(H, H^{'}) = \mathbb{E}_{h, h_{pos}^{'} \sim p(H, H'), h_{neg}^{'} \sim p(h')}[\log \frac{\exp \phi(h, h_{pos}^{'})}{\sum_{h_{neg}^{'}} \exp \phi(h, h_{neg}^{'})}]. \tag{6}$$

Based on InfoNCE, we leverage to $\mathcal{L}_1(i) = I_{NCE}(H_i^{'}, H_i^{''})$ represent $\sum_i U_i + R + S$, and utilize $\mathcal{L}_2(i) = \frac{1}{2}(I_{NCE}(H_i, H_i^{'}) + I_{NCE}(H_i, H_i^{''}))$ to denote $R + \sum_i U_i$. Then, the total training objective can be defined as follows:

$$\mathcal{L}_a = -\sum_{i=1}^{2} [\mathcal{L}_1(i) + \mathcal{L}_2(i)], \tag{7}$$

Clustering Loss. For more reliable clustering performance, we apply K-Means on Z to derive the pseudo-labels $\hat{y}$. Then, we define soft clustering assignment q and generate target distribution p (Zhang et al., 2026d):

$$q_{ij} = \frac{(1 + ||z_i - \mathbf{c}_j||^2)^{-1}}{\sum_{k=1}^{C} (1 + ||z_i - \mathbf{c}_k||^2)^{-1}}, p_{ij} = \frac{(q_{ij} / \sum_{i=1}^{N} q_{ij})^2}{\sum_{k=1}^{C} q_{ik} / \sum_{i=1}^{N} q_{ik}}. \tag{8}$$

Following Zhao et al. (2025), we finally employ cross-entropy loss to steer the optimization of clusters:

$$\mathcal{L}_c = -\sum_{b=1}^{B} q_b \log(p_b). \tag{9}$$

This function helps to improve the quality of the soft clustering assignments, enhancing the reliability of the clustering observations.

Total Loss. Given the mutual information estimator and the clustering refinement, our total loss can be formulated as follows:

$$\mathcal{L} = \gamma \mathcal{L}_a + \mathcal{L}_c, \tag{10}$$

where γ is a hyperparameter to weight the contributions of $\mathcal{L}_a$.

Inference. With frozen parameters, we directly utilize H_1 and H_2 to generate integrated representations, and then implement clustering using the K-Means algorithm. Section G shows pseudo-code.

5 Experiments

5.1 Experimental Settings

Datasets. For node-level clustering, we introduce six datasets, including ACM, DBLP, IMDB, Texas, Chameleon, and Wisconsin (Ren et al., 2025; Wang et al., 2026b). For graph-level clustering, we introduce BZR, NCI1, ENZYMES, COLLAB, IMDB-BINARY, and REDDIT-M-12K (Wang et al., 2026a). More details about the datasets are in Section B.

Table 1: Clustering performance for node-level clustering. The best observations are highlighted **in bold**. Results are reported as $mean_{\pm std}$ in %.

Metric	Dataset	VGMGC	NGCE	TOTF	ASIL	EGTFC	SynC	UMGC-Net	GCCR-VGAE	SR-SPG	FVD	FairGC	PIDGC
NMI	ACM	$68.3_{\pm0.8}$	$71.8_{\pm1.2}$	$66.5_{\pm0.6}$	$72.9_{\pm1.5}$	$58.1_{\pm0.9}$	$70.4_{\pm1.1}$	$73.6_{\pm1.7}$	$70.6_{\pm0.7}$	$73.4_{\pm1.3}$	$75.1_{\pm1.0}$	$70.9_{\pm1.6}$	$\mathbf{77.2}_{\pm1.4}$
ARI		$73.5_{\pm1.1}$	$83.2_{\pm0.7}$	$61.7_{\pm1.8}$	$68.4_{\pm0.9}$	$64.9_{\pm1.4}$	$67.1_{\pm0.6}$	$82.6_{\pm1.6}$	$78.8_{\pm1.2}$	$69.2_{\pm0.8}$	$83.5_{\pm1.5}$	$72.4_{\pm1.0}$	$\mathbf{89.3}_{\pm1.3}$
ACC		$84.1_{\pm0.9}$	$86.5_{\pm1.4}$	$83.7_{\pm1.1}$	$87.2_{\pm0.6}$	$79.4_{\pm1.7}$	$87.8_{\pm1.2}$	$88.1_{\pm0.8}$	$85.3_{\pm1.5}$	$87.5_{\pm1.0}$	$88.6_{\pm0.7}$	$87.9_{\pm1.6}$	$\mathbf{91.8}_{\pm1.1}$
NMI	DBLP	$62.8_{\pm1.3}$	$69.6_{\pm0.8}$	$72.8_{\pm1.6}$	$71.1_{\pm1.1}$	$70.9_{\pm0.7}$	$76.3_{\pm1.4}$	$68.5_{\pm1.0}$	$72.5_{\pm1.8}$	$69.3_{\pm0.9}$	$75.7_{\pm1.2}$	$70.2_{\pm1.5}$	$\mathbf{79.2}_{\pm1.7}$
ARI		$70.6_{\pm1.2}$	$78.7_{\pm0.6}$	$83.2_{\pm1.5}$	$82.4_{\pm1.0}$	$80.6_{\pm1.7}$	$83.9_{\pm0.9}$	$77.1_{\pm1.3}$	$82.8_{\pm1.1}$	$79.2_{\pm0.8}$	$81.3_{\pm1.6}$	$80.0_{\pm1.4}$	$\mathbf{87.3}_{\pm1.8}$
ACC		$75.9_{\pm1.1}$	$80.1_{\pm1.5}$	$78.5_{\pm0.7}$	$83.2_{\pm1.3}$	$85.3_{\pm1.6}$	$88.7_{\pm1.0}$	$80.6_{\pm1.8}$	$83.4_{\pm0.9}$	$84.7_{\pm1.4}$	$86.4_{\pm0.8}$	$85.2_{\pm1.2}$	$\mathbf{91.0}_{\pm1.0}$
NMI	IMDB	$4.7_{\pm0.3}$	$5.7_{\pm0.2}$	$0.5_{\pm0.1}$	$5.4_{\pm0.4}$	$1.3_{\pm0.0}$	$1.0_{\pm0.0}$	$4.5_{\pm0.2}$	$5.8_{\pm0.5}$	$6.4_{\pm0.3}$	$3.2_{\pm0.1}$	$7.8_{\pm0.4}$	$\mathbf{9.7}_{\pm0.5}$
ARI		$10.4_{\pm0.7}$	$12.2_{\pm1.2}$	$5.7_{\pm0.4}$	$3.9_{\pm0.3}$	$1.8_{\pm0.0}$	$2.3_{\pm0.2}$	$4.1_{\pm0.5}$	$6.3_{\pm0.1}$	$8.2_{\pm0.4}$	$9.6_{\pm0.3}$	$3.9_{\pm0.2}$	$\mathbf{18.6}_{\pm1.5}$
ACC		$42.1_{\pm1.2}$	$54.7_{\pm0.8}$	$36.5_{\pm1.5}$	$42.3_{\pm1.0}$	$39.8_{\pm0.6}$	$37.2_{\pm1.7}$	$43.9_{\pm1.3}$	$45.1_{\pm0.9}$	$48.6_{\pm1.6}$	$50.7_{\pm1.1}$	$47.3_{\pm0.7}$	$\mathbf{60.7}_{\pm0.7}$
NMI	Texas	$22.4_{\pm1.0}$	$47.7_{\pm1.6}$	$54.8_{\pm0.9}$	$50.2_{\pm1.3}$	$37.5_{\pm1.7}$	$28.9_{\pm0.8}$	$48.6_{\pm1.2}$	$52.5_{\pm1.4}$	$39.8_{\pm0.7}$	$52.1_{\pm1.5}$	$59.5_{\pm1.1}$	$\mathbf{60.3}_{\pm1.8}$
ARI		$19.8_{\pm0.9}$	$50.4_{\pm1.4}$	$34.6_{\pm1.2}$	$10.1_{\pm0.5}$	$15.9_{\pm1.6}$	$33.2_{\pm0.7}$	$48.7_{\pm1.1}$	$12.9_{\pm0.8}$	$40.6_{\pm1.3}$	$9.8_{\pm0.4}$	$46.3_{\pm1.5}$	$\mathbf{65.5}_{\pm1.7}$
ACC		$47.2_{\pm1.1}$	$74.8_{\pm0.6}$	$71.5_{\pm1.5}$	$68.9_{\pm1.2}$	$59.4_{\pm1.7}$	$44.7_{\pm0.8}$	$66.1_{\pm1.4}$	$70.2_{\pm0.9}$	$63.9_{\pm1.6}$	$68.8_{\pm1.0}$	$72.1_{\pm1.3}$	$\mathbf{80.3}_{\pm0.6}$
NMI	Chameleon	$20.3_{\pm1.2}$	$20.6_{\pm0.8}$	$7.4_{\pm0.3}$	$10.8_{\pm0.6}$	$8.7_{\pm0.4}$	$6.2_{\pm0.2}$	$24.9_{\pm1.5}$	$7.6_{\pm0.5}$	$25.3_{\pm1.0}$	$9.9_{\pm0.4}$	$25.1_{\pm1.3}$	$\mathbf{31.6}_{\pm1.7}$
ARI		$17.1_{\pm1.0}$	$19.7_{\pm1.4}$	$5.5_{\pm0.2}$	$8.6_{\pm0.5}$	$3.2_{\pm0.3}$	$6.8_{\pm0.1}$	$19.4_{\pm1.6}$	$4.7_{\pm0.4}$	$22.3_{\pm1.2}$	$8.4_{\pm0.3}$	$20.1_{\pm0.9}$	$\mathbf{25.3}_{\pm1.5}$
ACC		$45.6_{\pm1.3}$	$42.9_{\pm0.9}$	$33.1_{\pm1.6}$	$41.2_{\pm1.1}$	$36.7_{\pm1.5}$	$37.5_{\pm0.7}$	$43.4_{\pm1.2}$	$35.9_{\pm1.8}$	$44.1_{\pm1.0}$	$40.7_{\pm0.8}$	$45.3_{\pm1.4}$	$\mathbf{54.8}_{\pm1.2}$
NMI	Wisconsin	$6.4_{\pm0.3}$	$42.6_{\pm1.5}$	$7.1_{\pm0.2}$	$13.4_{\pm0.8}$	$8.7_{\pm0.4}$	$19.5_{\pm1.2}$	$7.8_{\pm0.5}$	$39.7_{\pm1.6}$	$25.5_{\pm1.0}$	$4.9_{\pm0.1}$	$38.6_{\pm1.3}$	$\mathbf{54.2}_{\pm1.8}$
ARI		$5.1_{\pm0.2}$	$46.9_{\pm1.4}$	$10.6_{\pm0.6}$	$9.3_{\pm0.4}$	$5.8_{\pm0.3}$	$28.1_{\pm1.7}$	$9.4_{\pm0.5}$	$17.3_{\pm1.1}$	$15.2_{\pm0.9}$	$7.2_{\pm0.1}$	$18.8_{\pm1.5}$	$\mathbf{58.7}_{\pm1.6}$
ACC		$33.8_{\pm1.2}$	$70.6_{\pm0.7}$	$35.4_{\pm1.6}$	$50.1_{\pm1.0}$	$49.2_{\pm1.3}$	$57.9_{\pm0.9}$	$42.3_{\pm1.5}$	$66.7_{\pm1.1}$	$61.9_{\pm1.4}$	$56.8_{\pm0.8}$	$70.3_{\pm1.7}$	$\mathbf{78.5}_{\pm0.9}$

Baselines. For node-level clustering methods, we introduce the following baselines: VGMGC (Chen et al., 2025), NGCE (Ren et al., 2025), TOTF (Li et al., 2025), ASIL (Sun et al., 2026a), EGTFC (Wang et al., 2026b), SynC (Ding et al., 2026a), UMGC-Net (Peng et al., 2026), GCCR-VGAE (Aghayengejeh et al., 2026), SR-SPG (Ding et al., 2026b), FVD (Cheng et al., 2026), and FairGC (Zhang et al., 2026b). For graph-level clustering methods, we introduce the following baselines: GLCC (Ju et al., 2023), MGSN (Han et al., 2025b) DCGLC (Cai et al., 2024), DBGCN (Han et al., 2025a), ANGC (Wang et al., 2026a), AMMGC (Zhao et al., 2025), AMGC (Yang et al., 2025), SCGC-DLA (Zhang et al., 2026a), BOGK (Ouyang et al., 2026), DNA-CGC (Zhao et al., 2026), UGCN (Han et al., 2026), and SMGCN (Hu et al., 2026). More details are in Section C.

Implementations. The hidden dimension d' is set to 64. Hyperparameters γ and K are, respectively, set to 0.5 and 5. The training epoch is set to 200. The learning rate is 1e-4. All methods are executed in a Python 3.9.12 environment, utilizing a single NVIDIA A40 GPU, 40GB of RAM, and a 2.60GHz Xeon (R) Gold 6240 CPU. We utilize two metrics to evaluate the performance: AUROC and AUPRC. Each experiment is run 5 times with 5 different random seeds, and the average performance is recorded with standard deviation.

Table 2: Clustering performance for graph-level clustering. The best observations are highlighted **in bold**.

Metric	Dataset	GLCC	MGSN	DCGLC	DBGCN	ANGC	AMMGC	AMGC	SCGC-DLA	BOGK	DNA-CGC	SMGCN	PIDGC
NMI	BZR	$2.3_{\pm 0.2}$	$4.8_{\pm 0.3}$	$5.6_{\pm 0.4}$	$2.2_{\pm 0.1}$	$9.1_{\pm 0.5}$	$1.5_{\pm 0.0}$	$6.4_{\pm 0.3}$	$4.3_{\pm 0.2}$	$3.7_{\pm 0.4}$	$5.9_{\pm 0.1}$	$6.8_{\pm 0.5}$	$\mathbf{13.7}_{\pm 1.2}$
ARI		$1.1_{\pm 0.0}$	$4.7_{\pm 0.3}$	$5.8_{\pm 0.2}$	$1.9_{\pm 0.0}$	$8.4_{\pm 0.4}$	$1.6_{\pm 0.0}$	$6.1_{\pm 0.5}$	$3.2_{\pm 0.1}$	$2.8_{\pm 0.3}$	$4.6_{\pm 0.2}$	$5.3_{\pm 0.4}$	$\mathbf{10.5}_{\pm 0.8}$
ACC		$58.9_{\pm 1.3}$	$68.9_{\pm 1.5}$	$73.1_{\pm 1.1}$	$65.3_{\pm 1.8}$	$75.6_{\pm 0.9}$	$74.2_{\pm 1.6}$	$70.8_{\pm 1.2}$	$66.4_{\pm 1.4}$	$70.6_{\pm 0.7}$	$64.9_{\pm 1.9}$	$73.3_{\pm 1.0}$	$\mathbf{79.5}_{\pm 1.3}$
NMI	NCI1	$2.1_{\pm 0.1}$	$4.4_{\pm 0.4}$	$9.8_{\pm 0.5}$	$1.8_{\pm 0.0}$	$4.8_{\pm 0.2}$	$5.2_{\pm 0.3}$	$7.1_{\pm 0.5}$	$6.3_{\pm 0.4}$	$6.7_{\pm 0.1}$	$5.8_{\pm 0.3}$	$8.2_{\pm 0.2}$	$\mathbf{12.4}_{\pm 1.5}$
ARI		$1.1_{\pm 0.0}$	$1.7_{\pm 0.0}$	$5.1_{\pm 0.3}$	$1.5_{\pm 0.0}$	$2.2_{\pm 0.1}$	$3.8_{\pm 0.4}$	$5.4_{\pm 0.2}$	$4.6_{\pm 0.5}$	$3.3_{\pm 0.3}$	$3.9_{\pm 0.1}$	$5.7_{\pm 0.4}$	$\mathbf{7.9}_{\pm 0.4}$
ACC		$50.8_{\pm 1.6}$	$58.8_{\pm 1.2}$	$66.5_{\pm 1.4}$	$54.1_{\pm 1.9}$	$60.7_{\pm 1.1}$	$61.4_{\pm 0.8}$	$65.9_{\pm 1.7}$	$71.4_{\pm 1.3}$	$63.2_{\pm 1.5}$	$68.9_{\pm 0.9}$	$66.1_{\pm 1.0}$	$\mathbf{72.6}_{\pm 1.1}$
NMI	ENZYMES	$2.1_{\pm 0.2}$	$5.2_{\pm 0.3}$	$4.7_{\pm 0.1}$	$3.1_{\pm 0.4}$	$6.6_{\pm 0.5}$	$4.9_{\pm 0.2}$	$8.7_{\pm 0.4}$	$5.4_{\pm 0.3}$	$4.8_{\pm 0.1}$	$6.1_{\pm 0.5}$	$7.3_{\pm 0.2}$	$\mathbf{9.8}_{\pm 0.3}$
ARI		$0.3_{\pm 0.0}$	$1.2_{\pm 0.0}$	$1.4_{\pm 0.0}$	$0.1_{\pm 0.0}$	$2.8_{\pm 0.1}$	$0.7_{\pm 0.0}$	$1.3_{\pm 0.0}$	$2.6_{\pm 0.2}$	$1.9_{\pm 0.0}$	$2.4_{\pm 0.3}$	$3.1_{\pm 0.4}$	$\mathbf{4.5}_{\pm 0.2}$
ACC		$15.6_{\pm 1.3}$	$24.7_{\pm 1.1}$	$19.6_{\pm 1.6}$	$25.8_{\pm 1.4}$	$22.4_{\pm 0.9}$	$20.1_{\pm 1.7}$	$23.9_{\pm 1.2}$	$21.3_{\pm 1.8}$	$18.7_{\pm 1.5}$	$25.3_{\pm 1.0}$	$21.9_{\pm 1.6}$	$\mathbf{30.5}_{\pm 0.6}$
NMI	COLLAB	$3.4_{\pm 0.3}$	$25.6_{\pm 1.7}$	$24.7_{\pm 1.5}$	$5.8_{\pm 0.4}$	$11.3_{\pm 0.6}$	$7.4_{\pm 0.2}$	$27.1_{\pm 1.9}$	$9.2_{\pm 0.5}$	$10.6_{\pm 0.8}$	$12.8_{\pm 1.1}$	$18.4_{\pm 1.3}$	$\mathbf{36.6}_{\pm 1.9}$
ARI		$8.3_{\pm 0.5}$	$15.7_{\pm 1.2}$	$18.8_{\pm 1.6}$	$3.5_{\pm 0.2}$	$8.1_{\pm 0.4}$	$4.9_{\pm 0.3}$	$12.2_{\pm 0.7}$	$6.3_{\pm 0.1}$	$7.9_{\pm 0.5}$	$9.6_{\pm 0.4}$	$11.8_{\pm 1.0}$	$\mathbf{28.6}_{\pm 1.2}$
ACC		$53.7_{\pm 1.4}$	$62.7_{\pm 1.8}$	$63.9_{\pm 1.1}$	$55.1_{\pm 1.6}$	$60.6_{\pm 0.9}$	$57.3_{\pm 1.3}$	$64.8_{\pm 1.5}$	$58.4_{\pm 1.7}$	$60.2_{\pm 1.0}$	$59.4_{\pm 1.2}$	$60.3_{\pm 1.9}$	$\mathbf{72.9}_{\pm 1.0}$
NMI	IMDB-BINARY	$1.6_{\pm 0.0}$	$14.1_{\pm 1.3}$	$4.3_{\pm 0.2}$	$10.9_{\pm 0.7}$	$5.6_{\pm 0.4}$	$12.7_{\pm 1.1}$	$6.2_{\pm 0.5}$	$7.4_{\pm 0.3}$	$8.1_{\pm 0.5}$	$3.8_{\pm 0.1}$	$9.6_{\pm 0.2}$	$\mathbf{21.4}_{\pm 1.7}$
ARI		$1.4_{\pm 0.0}$	$15.9_{\pm 1.5}$	$2.1_{\pm 0.1}$	$9.3_{\pm 0.5}$	$7.6_{\pm 0.3}$	$15.8_{\pm 1.2}$	$6.7_{\pm 0.4}$	$4.9_{\pm 0.2}$	$5.7_{\pm 0.5}$	$5.2_{\pm 0.1}$	$8.3_{\pm 0.4}$	$\mathbf{24.5}_{\pm 1.3}$
ACC		$51.1_{\pm 1.9}$	$67.2_{\pm 1.2}$	$55.1_{\pm 1.5}$	$60.7_{\pm 1.4}$	$54.3_{\pm 1.7}$	$69.6_{\pm 1.1}$	$58.8_{\pm 1.6}$	$60.3_{\pm 1.0}$	$62.8_{\pm 1.8}$	$53.5_{\pm 1.3}$	$66.9_{\pm 0.8}$	$\mathbf{75.9}_{\pm 0.8}$
NMI	REDDIT-M-12K	$1.2_{\pm 0.0}$	$1.6_{\pm 0.0}$	$6.3_{\pm 0.4}$	$10.1_{\pm 0.6}$	$6.9_{\pm 0.3}$	$9.7_{\pm 0.5}$	$12.4_{\pm 0.9}$	$7.8_{\pm 0.2}$	$5.9_{\pm 0.4}$	$8.6_{\pm 0.1}$	$11.2_{\pm 0.8}$	$\mathbf{19.2}_{\pm 1.4}$
ARI		$0.6_{\pm 0.0}$	$2.1_{\pm 0.1}$	$1.8_{\pm 0.0}$	$2.9_{\pm 0.3}$	$3.2_{\pm 0.2}$	$4.7_{\pm 0.5}$	$7.5_{\pm 0.4}$	$8.4_{\pm 0.3}$	$3.8_{\pm 0.1}$	$5.1_{\pm 0.2}$	$6.7_{\pm 0.5}$	$\mathbf{10.4}_{\pm 0.6}$
ACC		$9.1_{\pm 0.5}$	$11.2_{\pm 0.7}$	$14.6_{\pm 1.1}$	$17.9_{\pm 1.4}$	$12.3_{\pm 0.9}$	$19.7_{\pm 1.6}$	$23.1_{\pm 1.2}$	$22.8_{\pm 1.5}$	$20.5_{\pm 1.3}$	$18.8_{\pm 1.0}$	$19.7_{\pm 1.7}$	$\mathbf{30.7}_{\pm 0.6}$

Table 3: Experimental results of ablation on the ACC (%) metric of PIDGC.

Method	ACM	DBLP	IMDB	Texas	Chameleon	Wisconsin	BZR	NCI1	ENZYMES	COLLAB	IMDB-B	REDDIT-M12K
w/o $\mathcal{L}_a$	$79.4_{\pm 1.3}$	$82.3_{\pm 0.7}$	$45.5_{\pm 1.6}$	$73.5_{\pm 1.1}$	$36.5_{\pm 0.9}$	$69.1_{\pm 1.8}$	$64.8_{\pm 1.4}$	$61.4_{\pm 0.6}$	$22.9_{\pm 1.2}$	$56.4_{\pm 1.7}$	$70.6_{\pm 1.0}$	$11.5_{\pm 0.5}$
w/o $\mathcal{L}_1$	$84.6_{\pm 0.8}$	$77.5_{\pm 1.5}$	$50.9_{\pm 1.2}$	$63.2_{\pm 1.9}$	$49.2_{\pm 0.7}$	$65.6_{\pm 1.3}$	$73.2_{\pm 1.6}$	$57.2_{\pm 0.9}$	$11.8_{\pm 0.6}$	$64.8_{\pm 1.4}$	$64.0_{\pm 1.1}$	$24.2_{\pm 1.8}$
w/o $\mathcal{L}_2$	$72.7_{\pm 1.7}$	$85.6_{\pm 1.1}$	$48.1_{\pm 0.5}$	$70.0_{\pm 1.4}$	$39.0_{\pm 1.8}$	$70.8_{\pm 0.9}$	$61.3_{\pm 1.2}$	$63.7_{\pm 1.6}$	$25.4_{\pm 0.7}$	$59.2_{\pm 1.3}$	$59.1_{\pm 1.5}$	$19.4_{\pm 1.0}$
$\mathcal{L}$	$\mathbf{91.8}_{\pm 1.1}$	$\mathbf{91.0}_{\pm 1.0}$	$\mathbf{60.7}_{\pm 0.7}$	$\mathbf{80.3}_{\pm 0.6}$	$\mathbf{54.8}_{\pm 1.2}$	$\mathbf{78.5}_{\pm 0.9}$	$\mathbf{79.5}_{\pm 1.3}$	$\mathbf{72.6}_{\pm 1.1}$	$\mathbf{30.5}_{\pm 0.6}$	$\mathbf{72.9}_{\pm 1.0}$	$\mathbf{75.9}_{\pm 0.8}$	$\mathbf{30.7}_{\pm 0.6}$

5.2 Experimental Analysis

Performance. Table 1 shows node-level clustering performance of PIDGC compared with state-of-the-art baselines. From the observations, we can see that PIDGC achieves the best performance on both node-level clustering tasks with respect to three metrics. This phenomenon can be attributed to the powerful ability of PIDGC to explore multiple interactions across different views and alleviate the strong assumption that cross-view information about the clustering objective is totally shared. For baselines, they perform competitively on some datasets, such as ACM, while exhibiting instability on others. Although they have the ability to mine different semantic views, they exhibit poorer performance on comprehensive evaluations compared with PIDGC. This may be attributed to the heterogeneity across datasets and the complicated distributions of some specific classes.

Table 2 shows the graph-level clustering performance of PIDGC compared with state-of-the-art baselines. PIDGC also achieves the best performance on the graph-level clustering task. We can see that performance varies significantly across datasets. For example, PIDGC achieves 79.5% ACC on BZR, while 30.5% on ENZYMES. This is because of the different cluster numbers across datasets and the difficulty of modeling discriminative feature characteristics.

Ablation. We implement four groups of ablation studies to validate the effectiveness of different components in PIDGC, including the loss function, fusion module, data augmentation, and distance measures.

(1) Loss Function. To check the utility of loss function, we verify $\mathcal{L}_a$, and the inside components $\mathcal{L}_1$ and $\mathcal{L}_2$. From the observations in Table 3, we can see that the ablation variants exhibit poorer performance than PIDGC, which demonstrates the effectiveness of $\mathcal{L}_1$ and $\mathcal{L}_2$. Specifically, $w/o\mathcal{L}_a$ exhibit the largest performance degradation, while $w/o\mathcal{L}_1$ and $w/o\mathcal{L}_2$ shows relatively poor performance than original training objective $\mathcal{L}$. This phenomenon indicates that different components have their unique contributions, and the synergistic effect of the two can help further improve performance.

(2) Fusion Module. To validate the effectiveness of using learnable parameters to fuse different views, we introduce shallow linear fusion and Transformer architecture as variants. From the observations in Table 4, we can see that using linear and Transformer architectures compromises the clustering performance. On the one hand, using linear fusion degrades the discrimination of the fused representations. On the other hand, the Transformer may introduce additional irrelevant information. However, using learnable parameters can more flexibly grasp view-specific contributions. More results are shown in Section D.

Table 4: The ablation performance of the fusion module experiment with respect to the node-level clustering.

Method	ACM			DBLP			IMDB			Texas			Chameleon			Wisconsin		
	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Linear	$70.5_{\pm1.2}$	$32.4_{\pm1.7}$	$83.7_{\pm0.9}$	$72.3_{\pm1.5}$	$15.6_{\pm0.8}$	$82.4_{\pm1.3}$	$7.8_{\pm0.3}$	$14.2_{\pm1.1}$	$52.6_{\pm1.6}$	$54.6_{\pm1.4}$	$36.1_{\pm0.7}$	$72.8_{\pm1.8}$	$26.8_{\pm1.0}$	$18.5_{\pm1.2}$	$46.9_{\pm1.5}$	$47.5_{\pm1.6}$	$45.8_{\pm0.9}$	$70.6_{\pm1.1}$
Transformer	$72.1_{\pm0.8}$	$29.8_{\pm1.3}$	$82.5_{\pm1.7}$	$69.8_{\pm1.1}$	$17.2_{\pm0.6}$	$80.9_{\pm1.4}$	$8.4_{\pm0.2}$	$13.1_{\pm0.9}$	$51.8_{\pm1.2}$	$52.9_{\pm1.5}$	$38.4_{\pm1.8}$	$74.2_{\pm0.7}$	$28.3_{\pm1.4}$	$17.4_{\pm0.5}$	$48.1_{\pm1.6}$	$45.9_{\pm1.0}$	$47.2_{\pm1.3}$	$72.4_{\pm0.8}$
PIDGC	$\mathbf{77.2}_{\pm1.4}$	$\mathbf{89.3}_{\pm1.3}$	$\mathbf{91.8}_{\pm1.1}$	$\mathbf{79.2}_{\pm1.7}$	$\mathbf{87.3}_{\pm1.8}$	$\mathbf{91.0}_{\pm1.0}$	$\mathbf{9.7}_{\pm0.5}$	$\mathbf{18.6}_{\pm1.5}$	$\mathbf{60.7}_{\pm0.7}$	$\mathbf{60.3}_{\pm1.8}$	$\mathbf{65.5}_{\pm1.7}$	$\mathbf{80.3}_{\pm0.6}$	$\mathbf{31.6}_{\pm1.7}$	$\mathbf{25.3}_{\pm1.5}$	$\mathbf{54.8}_{\pm1.2}$	$\mathbf{54.2}_{\pm1.8}$	$\mathbf{58.7}_{\pm1.6}$	$\mathbf{78.5}_{\pm0.9}$

Table 5: Experimental results of ablation on augmentation strategies of PIDGC on the ACC (%) metric. FM denotes feature mask, and ED denotes edge drops. By default, PIDGC utilizes FM for both views. T1 indicates the first augmentation strategy for each view, and T2 indicates the second augmentation strategy for each view.

T1	T2	ACM	DBLP	IMDB	Texas	Chameleon	Wisconsin	BZR	NCI1	ENZYMES	COLLAB	IMDB-B	REDDIT-M12K
ED	FM	$87.4_{\pm1.3}$	$88.1_{\pm0.9}$	$56.9_{\pm1.6}$	$76.2_{\pm1.1}$	$51.3_{\pm1.8}$	$74.8_{\pm0.7}$	$75.1_{\pm1.4}$	$68.9_{\pm1.0}$	$27.6_{\pm1.2}$	$69.4_{\pm0.8}$	$71.8_{\pm1.5}$	$27.9_{\pm1.7}$
ED	ED	$84.6_{\pm1.1}$	$85.7_{\pm1.5}$	$53.8_{\pm1.3}$	$73.1_{\pm0.8}$	$48.7_{\pm1.6}$	$71.6_{\pm1.2}$	$72.3_{\pm0.9}$	$65.4_{\pm1.7}$	$25.1_{\pm1.0}$	$66.8_{\pm1.4}$	$69.2_{\pm0.6}$	$25.3_{\pm1.8}$
FM	ED	$89.2_{\pm1.4}$	$89.6_{\pm1.0}$	$58.4_{\pm1.7}$	$78.5_{\pm1.2}$	$52.6_{\pm0.9}$	$76.3_{\pm1.5}$	$77.4_{\pm1.1}$	$70.1_{\pm1.8}$	$28.8_{\pm1.3}$	$70.9_{\pm0.7}$	$73.6_{\pm1.6}$	$28.6_{\pm1.9}$
FM	FM	$\mathbf{91.8}_{\pm1.4}$	$\mathbf{91.0}_{\pm1.0}$	$\mathbf{60.7}_{\pm0.7}$	$\mathbf{80.3}_{\pm0.6}$	$\mathbf{54.8}_{\pm1.2}$	$\mathbf{78.5}_{\pm0.9}$	$\mathbf{79.5}_{\pm1.3}$	$\mathbf{72.6}_{\pm1.1}$	$\mathbf{30.5}_{\pm0.6}$	$\mathbf{72.9}_{\pm1.0}$	$\mathbf{75.9}_{\pm0.8}$	$\mathbf{30.7}_{\pm0.6}$

(3) Data Augmentation. We introduce additional data augmentation strategies to explore the influence on the clustering performance. From the observations in Table 5, we can see that using different augmentation strategies results in slight performance fluctuations. Specifically, using FM and FM achieves comprehensively superior performance. Besides, using ED and FM also exhibits competitive performance. However, using ED and ED leads to relatively poor ACC, such as on the ACM, DBLP, and COLLAB datasets. Therefore, the best augmentation strategy is FM.

(4) Distance Measures. This section aims to validate the influence of different distance measures on clustering performance. We introduce three distance measures to generate a complementary graph view, including Wasserstein distance, Jaccard similarity, and Euclidean distance. The observations on node-level clustering are shown in Table 6. Using Euclidean distance results in significant performance degradation. The reason is that Euclidean similarity may exhibit poorer ability to evaluate the semantics between nodes, thereby generating a complementary graph view with less discrimination. However, using Wasserstein or Jaccard similarity to generate a complementary graph view exhibits competitive performance compared to using cosine similarity, even achieving better performance on some datasets. More results on graph-level clustering are shown in Section E.

Table 6: The results of distance-measure ablation with respect to ACC (%).

Method	ACM	DBLP	IMDB	Texas	Chameleon	Wisconsin
Wasserstein	$83.6_{\pm1.2}$	$89.0_{\pm0.9}$	$56.1_{\pm1.4}$	$76.5_{\pm1.1}$	$51.9_{\pm1.6}$	$\mathbf{78.7}_{\pm0.8}$
Jaccard	$87.8_{\pm1.0}$	$\mathbf{92.3}_{\pm0.7}$	$57.2_{\pm1.3}$	$73.9_{\pm1.5}$	$50.6_{\pm1.2}$	$77.9_{\pm1.0}$
Euclidean	$80.2_{\pm1.5}$	$83.9_{\pm1.1}$	$53.4_{\pm1.7}$	$71.3_{\pm1.3}$	$49.7_{\pm1.4}$	$66.8_{\pm1.6}$
PIDGC	$\mathbf{91.8}_{\pm1.4}$	$91.0_{\pm1.0}$	$\mathbf{60.7}_{\pm0.7}$	$\mathbf{80.3}_{\pm0.6}$	$\mathbf{54.8}_{\pm1.2}$	$78.5_{\pm0.9}$

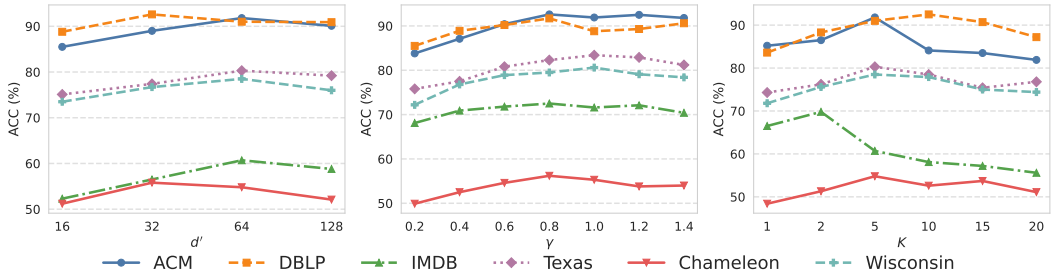


Figure 2: Sensitivity experiments on the node-level clustering task.

Sensitivity. We testify the sensitivity of hyperparameters, including d, γ, and K. The observations of the node-level clustering task are shown in Fig. 2. For d', we can see that the model has relatively poor performance on the Chameleon and Wisconsin datasets compared with other datasets. This is because the hidden size may influence the expressive ability of representation spaces. Increasing γ directly results in an improvement in performance because the utility of $\mathcal{L}_a$ significantly enlarges.

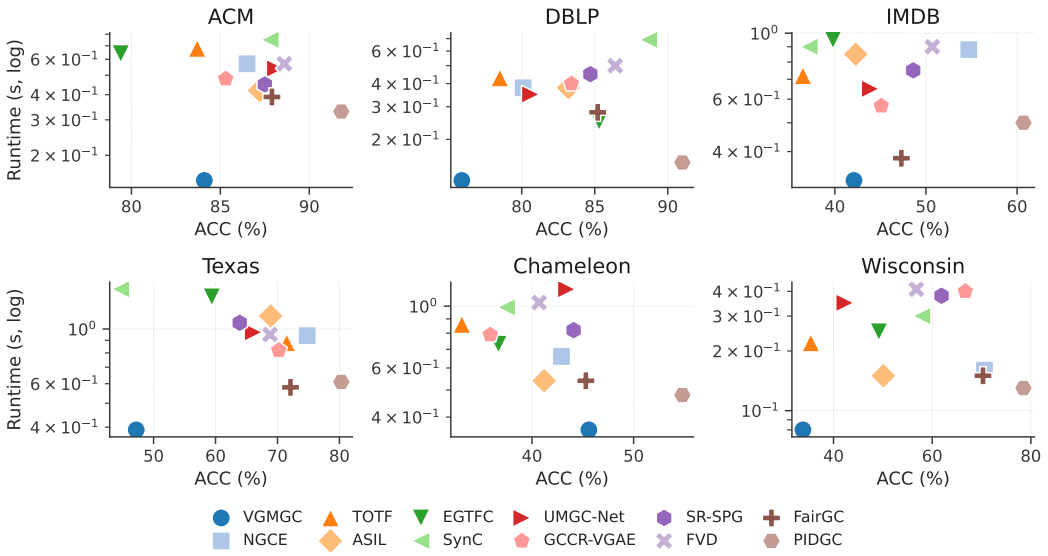


Figure 3: The efficiency experiments on the node-level clustering with respect to PIDGC and baselines.

However, the performance tends to be stable when the value exceeds a certain threshold. For K, it controls the complexity of the complementary graph view and determines the semantic quality. When K is set to a relatively small value, it can result in too few semantic nodes being associated with a node, leading to impoverished semantic information. When K is set to a relatively large value, the semantics tend to be richer, but redundant and interference information may also be introduced.

Efficiency. We conduct efficiency experiments to validate PIDGC's computational cost compared with baselines. As shown in Fig. 3, PIDGC achieves clear superiority over most baselines on most datasets. This phenomenon is evident on the ACM, DBLP, IMDB, and Chemeleon datasets. However, on the Texas and Wisconsin datasets, some baselines achieve competitive performance, such as FairGC. This is because the number of nodes in these datasets is relatively small. Notably, VGMGC exhibits superior efficiency over all models, which is attributed to its lightweight model architecture and the low-cost learning strategy.

6 Conclusion

This paper studies the multi-view graph clustering problem and proposes a multi-view graph clustering framework based on partial information decomposition (PIDGC) to alleviate the strong assumption that cross-view information about the clustering objective is entirely shared. PIDGC is a multi-view lightweight contrastive learning paradigm that combines view-specific representations into a unified latent space for clustering. Unlike existing graph clustering methods, PIDGC is designed to explore multiple interactions across views beyond redundancy via partial information decomposition. We provide theoretical support for learning different types of interactions via contrastive learning. Extensive experiments demonstrate the effectiveness of PIDGC.

The present work leaves ample room for further inquiry into multi-view graph clustering, particularly for developing augmentation schemes that retain cluster-relevant semantics across multiple views and go beyond perturbations confined to a single view. A broader discussion of limitations and promising avenues is provided in Section H. In sum, the proposed framework combines simplicity with flexibility, making it a compelling option for learning deep representations in multi-view graph clustering across diverse domains.

AI USE STATEMENT

During manuscript preparation, the authors used ChatGPT solely for grammatical correction and English-language polishing. The tool was not used for research design, methodological development, code generation, experimentation, data analysis, or the generation of scientific results or conclusions. The authors take full responsibility for the content, accuracy, and integrity of the manuscript.

ETHICS STATEMENT

This study does not involve human subjects, practices related to dataset releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues (e.g., IRB, documentation, research ethics).

REPRODUCIBILITY STATEMENT

Sections 3 and 4 define the problem, the proposed framework, and the corresponding objective. The supplement specifies more details about experimental configuration and results. All evaluated datasets are public and cited; no private data are used.

REFERENCES

Nazila Pourhaji Aghayengejeh, M. A. Balafar, Jafar Tanha, and Aryaz Baradarani. Graph-to-cluster contrastive ratio variational graph autoencoder for deep graph clustering. *Appl. Soft Comput.*, 203:116240, 2026. doi: 10.1016/J.ASOC.2026.116240.

Mohamed Mahmoud Amar, Nairouz Mrabah, Mohamed Bouguessa, and Abdoulaye Baniré Diallo. A unified perspective for learning graph representations across multi-level abstractions. *IEEE Trans. Knowl. Data Eng.*, 38(7):4076–4092, 2026. doi: 10.1109/TKDE.2026.3681811.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014. doi: 10.3390/E16042161.

Jinyu Cai, Yunhe Zhang, Jicong Fan, Yali Du, and Wenzhong Guo. Dual contrastive graph-level clustering with multiple cluster perspectives alignment. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3770–3779, 2024.

Jianpeng Chen, Yawen Ling, Jie Xu, Yazhou Ren, Shudong Huang, Xiaorong Pu, Zhifeng Hao, Philip S. Yu, and Lifang He. Variational graph generator for multiview graph clustering. *IEEE Trans. Neural Networks Learn. Syst.*, 36(6):11078–11091, 2025. doi: 10.1109/TNNLS.2024.3524205.

Junwei Cheng, Ke Liang, Pengxing Feng, Weixiong Liu, Yong Tang, and Chaobo He. Clustering diffusion model with frequency-signal modulation for variational graph autoencoders. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(2):1503–1520, 2026. doi: 10.1109/TPAMI.2025.3614385.

Shifei Ding, Benyu Wu, Xiao Xu, Ling Ding, and Xindong Wu. Sync: Synergistic boosting of structure and representation for deep graph clustering. *IEEE Trans. Neural Networks Learn. Syst.*, 37(6):2959–2968, 2026a. doi: 10.1109/TNNLS.2025.3643594.

Yuanfu Ding, Xiufang Xu, Sen Xu, Xuesheng Bian, Shanliang Yao, Naixuan Guo, Wei Wang, Tian Zhou, and Yuyang Shen. Contrastive graph clustering with structure-robust learning and stable prototype guidance. *Neural Networks*, 202:109056, 2026b. doi: 10.1016/J.NEUNET.2026.109056.

Renda Han, Wenxuan Tu, Junlong Wu, Wenxin Zhang, Jingxin Liu, Haotian Wang, Jieren Cheng, Huajie Lei, Guangzhen Yao, Lingren Wang, Mengfei Li, and Yu Li. Dual boost-driven graph-level clustering network. In *International Joint Conference on Neural Networks*, pp. 1–8, 2025a. doi: 10.1109/IJCNN64981.2025.11228488.

Renda Han, Guangzhen Yao, Mingyuan Li, Wenxin Zhang, Yu Li, Wen Xin, Huajie Lei, Mengfei Li, Zeyu Zhang, Chengze Du, and Yahe Tian. Multi-relation graph-kernel strengthen network for graph-level clustering. In *International Joint Conference on Neural Networks*, pp. 1–8, 2025b. doi: 10.1109/IJCNN64981.2025.11227596.

Renda Han, Xiaobao Wang, Longbiao Wang, Wenxin Zhang, Ronghao Fu, Kaiming Wang, Zeyu Zhang, and Kuntharrgyal Khysru. A unified graph clustering network. In *Proceedings of the ACM Web Conference 2026*, pp. 845–856, 2026. doi: 10.1145/3774904.3792266.

Tianyu Hu, Renda Han, Liu Mao, Jing Chen, and Xia Xie. Structure-missing graph-level clustering network. *Neural Networks*, 199:108682, 2026. doi: 10.1016/J.NEUNET.2026.108682.

Wei Ju, Yiyang Gu, Binqi Chen, Gongbo Sun, Yifang Qin, Xingyuming Liu, Xiao Luo, and Ming Zhang. GLCC: A general framework for graph-level clustering. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pp. 4391–4399, 2023. doi: 10.1609/AAAI.V37I4.25559.

Mengyao Li, Xu Zhou, Jiapeng Zhang, Zhibang Yang, Cen Chen, and Kenli Li. TOTF: missing-aware encoders for clustering on multi-view incomplete attributed graphs. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3054–3062, 2025. doi: 10.24963/IJCAI.2025/340.

Ke Liang, Lingyuan Meng, Hao Li, Jun Wang, Long Lan, Miaomiao Li, Xinwang Liu, and Huaimin Wang. From concrete to abstract: Multi-view clustering on relational knowledge. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(10):9043–9060, 2025. doi: 10.1109/TPAMI.2025.3582689.

Chuang Liu, Wenhang Yu, Kuang Gao, Xueqi Ma, Yibing Zhan, Jia Wu, Wenbin Hu, and Bo Du. Graph explicit pooling for graph-level representation learning. *Neural Networks*, 181:106790, 2025. doi: 10.1016/J.NEUNET.2024.106790.

Chao Ouyang, Haijun Zhang, and Jicong Fan. BOGK: bayesian optimization-driven graph kernel ensemble for graph-level clustering. *IEEE Trans. Knowl. Data Eng.*, 38(4):2183–2194, 2026. doi: 10.1109/TKDE.2026.3658514.

Bo Peng, Shaobo Bai, Jianjun Lei, Yuxuan Yao, Changqing Zhang, and Nam Ling. Uncertainty-aware multi-view graph clustering. *IEEE Trans. Multim.*, 28:2315–2324, 2026. doi: 10.1109/TMM.2026.3651061.

Ziming Quan, Penglei Wang, Danyang Wu, and Jin Xu. Unsupervised cross-view message passing method for multi-view graph clustering. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 1122–1131, 2025. doi: 10.1145/3746027.3754969.

Yazhou Ren, Junlong Ke, Zichen Wen, Tianyi Wu, Yang Yang, Xiaorong Pu, and Lifang He. Multi-view graph clustering via node-guided contrastive encoding. In *Forty-second International Conference on Machine Learning*, volume 267, 2025.

Li Sun, Zhenhao Huang, Yujie Wang, Hongbo Lv, Chunyang Liu, Hao Peng, and Philip S. Yu. ASIL: augmented structural information learning for deep graph clustering in hyperbolic space. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(6):6817–6834, 2026a. doi: 10.1109/TPAMI.2026.3661424.

Ruishuang Sun, Enguang Zuo, Xiaoyi Lv, Ziwei Yan, Chen Chen, and Cheng Chen. Towards superior graph representation: A multi-level cross-view approach integrating information bottleneck and contrastive learning. *Appl. Soft Comput.*, 197:115148, 2026b. doi: 10.1016/J.ASOC.2026.115148.

Zemin Tang, Qi Xiao, Yunchuan Qin, Xu Zhou, Joey Tianyi Zhou, and Kenli Li. Multi-view interactive representations for multimodal sentiment analysis. *IEEE Trans. Consumer Electron.*, 70 (1):4095–4107, 2024. doi: 10.1109/TCE.2024.3357480.

Jiaxin Wang, Wenxuan Tu, Lingren Wang, Jieren Cheng, and Yue Yang. Anchor-driven nyström for deep graph-level clustering. In *Fortieth AAAI Conference on Artificial Intelligence*, pp. 26260–26268, 2026a. doi: 10.1609/AAAI.V40I31.39830.

Penglei Wang, Jitao Lu, Danyang Wu, Rong Wang, and Feiping Nie. Enhance before fusion: Multi-view graph clustering with graph trend filter. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(6): 6297–6313, 2026b. doi: 10.1109/TPAMI.2026.3655829.

Chenyang Xie, Yingying Song, Song He, Xiaochen Bo, and Zhongnan Zhang. MMPCS: multi-view molecular pretraining based on consistency information and specific information. *Bioinform.*, 42 (2), 2026. doi: 10.1093/BIOINFORMATICS/BTAG028.

Jinbin Yang, Jinyu Cai, Yunhe Zhang, Sujia Huang, and Shiping Wang. Towards adaptive masked structural learning for graph-level clustering. *IEEE Trans. Netw. Sci. Eng.*, 12(3):2021–2032, 2025. doi: 10.1109/TNSE.2025.3543194.

Guang-Yu Zhang, Yan-Di Huang, Dong Huang, Chang-Dong Wang, Yang Liu, and Enbo Huang. Structure-preserving contrastive graph clustering with dual-channel label alignment. *Neural Networks*, 193:107954, 2026a. doi: 10.1016/J.NEUNET.2025.107954.

Haodong Zhang, Xinyue Wang, Tao Ren, Yifan Wang, Siyu Yi, Fanchun Meng, Zeyu Ma, Qingqing Long, and Wei Ju. Fairgc: Fostering individual and group fairness for deep graph clustering. In *Fortieth AAAI Conference on Artificial Intelligence*, pp. 28194–28202, 2026b. doi: 10.1609/AAAI.V40I33.40046.

Wenxin Zhang, Xi Xuan, Renda Han, Desheng Wu, Cuicui Luo, and Ljupco Kocarev. Graph clustering with scalable graph filters and view-specific semantic fusion. In *Database Systems for Advanced Applications*, volume 16536, pp. 83–97, 2026c. doi: 10.1007/978-981-92-0366-6_6.

Wenxin Zhang, Xi Xuan, Renda Han, Desheng Wu, Cuicui Luo, and Ljupco Kocarev. Graph clustering with scalable graph filters and view-specific semantic fusion. In *Database Systems for Advanced Applications - 31st International Conference, DASFAA 2026*, volume 16536, pp. 83–97, 2026d. doi: 10.1007/978-981-92-0366-6_6.

Bowen Zhao, Qianqian Wang, Zhengming Ding, and Quanxue Gao. Attribute-missing multi-view graph clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25832–25841, 2025. doi: 10.1109/CVPR52734.2025.02406.

Tianxiang Zhao, Youqing Wang, Shilong Xu, Tianchuan Yang, Junbin Gao, and Jipeng Guo. Dual-level noise augmentation for graph clustering with triplet-wise contrastive learning. *Pattern Recognit.*, 172:112463, 2026. doi: 10.1016/J.PATCOG.2025.112463.

A PROOF

Proof (Lemma 2) Given data processing inequalities for the Markov chains $X \to X' \to H_\theta$ and $H_\theta^{'} \to X \to H_\theta^{'}$, we have:

$$I(H_\theta, H_\theta^{'}) \leq I(X, H_\theta^{'}) \leq I(X, X^{'}). \tag{11}$$

Denote the function $f_\theta(X, A)$ is identity function, then the equality can be directly achieved.

Proof (Theorem 1) Given Lemma 1, we have:

$$\begin{aligned} I(X^{'}, Y) &= I(X^{'}, Y; X) + I(X^{'}, Y|X) && (12)\\ &= I(H_{\theta^*}^{'}; Y; X) && (13)\\ &= I(H_{\theta^*}^{'}; Y) - I(H_{\theta^*}^{'}; Y|X) && (14)\\ &= I(H_{\theta^*}^{'}; Y) && (15) \end{aligned}$$

where $H_{\theta^*}^{'} = f_{\theta^*}(f(X), A)$.

We define an augmentation strategy set $\mathcal{T}$ and an augmentation strategy $t \in \mathcal{T}$. Given $X^{'} = t(X)$, $H_{\theta^*}^{'} = H_i^{'}$ for view A_i and the chain rules of mutual information, we have:

$$\begin{aligned} I(H_{\theta^*}^{'}; Y) &= I(H_i^{'}; Y) && (16)\\ &= I(f(t(X), A_i); Y) && (17)\\ &= R + U_i. && (18) \end{aligned}$$

B DATASETS

Node-Level Clustering ACM is a paper network from the ACM database and is composed of two graphs: the co-paper network and the co-subject network; DBLP is a network sourced from the DBLP database, consists of three graphs: co-author, coconference, and co-term; IMDB is a movie network that originates from the IMDB dataset, including graphs of both co-actors and co-directors. Details are shown in Table 7.

Graph-Level Clustering All datasets are from the TUDatasets collection. Specifically, NCI1 is a small molecule dataset, ENZYMES is a bioinformatics dataset, COLLAB, IMDB-BINARY, REDDIT-M-12K are social network datasets. Details are shown in Table 8.

Table 7: Detailed statistics of the datasets for node-level clustering.

Dataset	# Nodes	# Dims	# Clusters
ACM	3025	1870	3
DBLP	4057	334	4
IMDB	4780	1232	3
Texas	183	1703	5
Chameleon	2277	2325	5
Wisconsin	251	1703	5

Table 8: Detailed statistics of the datasets for graph-level clustering.

Dataset	# Graphs	# Avg. Nodes	# Avg. Edges	# Clusters
BZR	405	35.75	38.36	2
NCI1	4110	41.22	43.45	2
ENZYMES	600	32.60	62.14	6
COLLAB	5000	74.49	2457.78	3
IMDB-BINARY	1000	19.77	96.53	2
REDDIT-M-12K	11929	391.41	456.89	11

C BASELINES

In this section, we introduce the details of baselines. We first explain the baselines for node-level clustering:

- VGMGC (Chen et al., 2025) infers a discrete consensus graph across views via variational inference, where the prior is derived from the observed graphs weighted by learnable per-view beliefs, and fuses the consensus and view-specific graphs with node features through a parameter-free message-passing encoder.
- NGCE (Ren et al., 2025) leverages node features to guide embedding generation, integrating homophilic and heterophilic information within a unified node-guided framework for multi-view graph clustering. However, it depends on the availability of node features, which inherently limits its direct applicability to datasets where such features are sparse or absent.
- ASIL (Sun et al., 2026a) offers a simple yet effective objective of augmented structural entropy to seamlessly integrate hyperbolic partitioning tree construction and contrastive learning. With a provable improvement in graph conductance, it achieves effective debiased graph clustering in linear complexity with respect to the graph size, without requiring the number of clusters.
- EGTFC (Wang et al., 2026b) explores the credible topological structure while accomplishing cross-view interactions via a credible cross-view graph enhancement module, and adapts the high-order Graph Trend Filter to reveal the inhomogeneities in graph smoothness levels. However, it primarily focuses on homogeneous graphs, and its scalability and robustness, for example on bipartite and noisy graphs, remain to be improved.

- TOTF (Li et al., 2025) is the first work to focus on multi-view attributed graph clustering with incomplete attributes. It trains missing-aware encoders first to capture view-specific information, and then freezes their parameters to discover common information available for clustering.
- UMGC-Net (Peng et al., 2026) estimates the uncertainty of different samples across different views in an unsupervised manner, thus learning multi-view features friendly to clustering. The trusted fusion of multi-view clustering distributions is effectively achieved via Dempster's combination rule.
- DBGCN (Han et al., 2025a) alternately promotes graph-level clustering and filters out interference information in a unified framework. It effectively extracts high-confidence graph-level features to generate a robust cluster distribution.
- ANGC (Wang et al., 2026a) computes graph similarities directly from node embeddings in a kernel space, bypassing pooling operations. Learnable graph Nystr"om anchors are introduced to support end-to-end optimization.
- AMMGC (Zhao et al., 2025) first imputes missing node attributes by leveraging neighborhood information through an adjacency matrix. It integrates a dual structure consistency module that aligns graph structures across multiple views, reducing redundancy and retaining key information, and introduces a high-confidence guidance module to improve the reliability of clustering.
- AMGC (Yang et al., 2025) jointly performs adaptive mask feature reconstruction to learn graph-level embeddings and conducts graph-level clustering within a unified framework. Clustering-oriented graph-level representations are also generated by maximizing the mutual information between the whole graph and substructure representations.
- DNA-CGC (Zhao et al., 2026) integrates low- and high-frequency graph signals while introducing Gaussian noise as beneficial perturbation, and constructs the triplet-wise contrastive pairs, mitigating the contrastive bias by preventing false negatives. However, although it performs well in static graph scenarios, it may still face challenges when dealing with highly sparse or dynamically evolving graphs.

We then introduce graph-level baselines:

- GLCC (Ju et al., 2023) constructs an adaptive affinity graph to explore instance- and cluster-level contrastive learning, and utilizes neighbor-aware pseudo-labels to reward the optimization of representation learning.
- MGSN (Han et al., 2025b) constructs multi-relation graphs to capture diverse semantic relationships between nodes and graphs, and employs graph kernel methods to extract graph similarity features. However, its scalability remains to be optimized for larger and more complex graph-level clustering tasks.
- DCGLC (Cai et al., 2024) introduces the Euclidean-based and subspace-based cluster heads to capture the cluster information from different cluster perspectives, and proposes a contrastive mechanism to align the cluster information derived from them.
- DBGCN (Han et al., 2025a) alternately promotes graph-level clustering and filters out interference information in a unified framework. High-confidence graph-level features are effectively extracted to generate a robust cluster distribution.
- ANGC (Wang et al., 2026a) computes graph similarities in a kernel space rather than relying on pooling operations. Learnable graph anchors are introduced to support end-to-end optimization of graph-level clustering.
- AMMGC (Zhao et al., 2025) first imputes missing node attributes by leveraging neighborhood information through an adjacency matrix. It integrates a dual structure consistency module that aligns graph structures across multiple views, reducing redundancy and retaining key information, and introduces a high-confidence guidance module to improve the reliability of clustering.
- AMGC (Yang et al., 2025) jointly performs adaptive mask feature reconstruction to learn graph-level embeddings and conducts graph-level clustering within a unified framework. Clustering-oriented graph-level representations are also generated by maximizing the mutual information between the whole graph and substructure representations.

Table 9: The ablation performance of fusion module with respect to graph-level clustering.

Method	BZR			NCI1			ENZYMES			COLLAB			IMDB-B			REDDIT-M12K		
	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
Linear	$8.6_{\pm 0.2}$	$6.4_{\pm 0.1}$	$65.2_{\pm 0.7}$	$7.8_{\pm 0.2}$	$4.9_{\pm 0.1}$	$59.7_{\pm 0.7}$	$6.1_{\pm 0.1}$	$2.8_{\pm 0.1}$	$20.9_{\pm 0.5}$	$22.4_{\pm 0.8}$	$17.5_{\pm 0.7}$	$61.1_{\pm 0.9}$	$8.2_{\pm 0.6}$	$15.3_{\pm 0.7}$	$62.5_{\pm 0.9}$	$11.9_{\pm 0.5}$	$2.6_{\pm 0.1}$	$19.2_{\pm 0.5}$
Transformer	$10.9_{\pm 0.6}$	$8.2_{\pm 0.2}$	$69.8_{\pm 0.8}$	$9.6_{\pm 0.3}$	$6.1_{\pm 0.2}$	$64.1_{\pm 0.8}$	$7.7_{\pm 0.2}$	$3.5_{\pm 0.1}$	$23.4_{\pm 0.6}$	$28.9_{\pm 0.9}$	$22.8_{\pm 0.8}$	$65.3_{\pm 1.0}$	$17.1_{\pm 0.7}$	$19.6_{\pm 0.8}$	$68.2_{\pm 1.0}$	$15.2_{\pm 0.7}$	$3.3_{\pm 0.1}$	$23.9_{\pm 0.6}$
PIDGC	$\mathbf{13.7}_{\pm 1.2}$	$\mathbf{10.5}_{\pm 0.8}$	$\mathbf{79.5}_{\pm 1.3}$	$\mathbf{12.4}_{\pm 1.5}$	$\mathbf{7.9}_{\pm 0.4}$	$\mathbf{72.6}_{\pm 1.1}$	$\mathbf{9.8}_{\pm 0.3}$	$\mathbf{4.5}_{\pm 0.2}$	$\mathbf{30.5}_{\pm 0.6}$	$\mathbf{36.6}_{\pm 1.9}$	$\mathbf{28.6}_{\pm 1.2}$	$\mathbf{72.9}_{\pm 1.0}$	$\mathbf{21.4}_{\pm 1.7}$	$\mathbf{24.5}_{\pm 1.3}$	$\mathbf{75.9}_{\pm 0.8}$	$\mathbf{19.2}_{\pm 1.4}$	$\mathbf{4.1}_{\pm 0.6}$	$\mathbf{30.7}_{\pm 0.6}$

Table 10: The results of distance measures ablation with respect to ACC (%) for graph-level clustering.

Method	BZR	NCI1	ENZYMES	COLLAB	IMDB-B	REDDIT-M12K
Wasserstein	$75.2_{\pm 0.8}$	$\mathbf{74.2}_{\pm 1.2}$	$28.1_{\pm 0.6}$	$70.1_{\pm 1.0}$	$72.8_{\pm 1.1}$	$28.9_{\pm 0.6}$
Jaccard	$76.8_{\pm 0.9}$	$69.5_{\pm 0.9}$	$29.3_{\pm 0.7}$	$\mathbf{73.5}_{\pm 0.9}$	$74.1_{\pm 0.8}$	$28.2_{\pm 0.7}$
Euclidean	$69.1_{\pm 1.1}$	$62.3_{\pm 1.4}$	$25.2_{\pm 0.5}$	$65.3_{\pm 1.3}$	$67.5_{\pm 1.2}$	$26.4_{\pm 0.9}$
PIDGC	$\mathbf{79.5}_{\pm 1.3}$	$72.6_{\pm 1.1}$	$\mathbf{30.5}_{\pm 0.6}$	$72.9_{\pm 1.0}$	$\mathbf{75.9}_{\pm 0.8}$	$\mathbf{30.7}_{\pm 0.6}$

- SCGC-DLA (Zhang et al., 2026a) generates two complementary augmented views via the low-pass and hybrid graph filters, and aligns high-confidence pseudo labels with latent semantic labels under a self-supervised learning scheme.
- BOGK (Ouyang et al., 2026) uses Bayesian optimization to integrate various graph kernels for graph-level clustering. The similarity matrices generated by different graph kernels are aggregated, and the aggregation weights and a thresholding parameter are automatically learned via maximizing internal cluster validity indices.
- DNA-CGC (Zhao et al., 2026) integrates low- and high-frequency graph signals while introducing Gaussian noise as beneficial perturbation, and constructs the triplet-wise contrastive pairs, mitigating the contrastive bias by preventing false negatives. However, although it performs well in static graph scenarios, it may still face challenges when dealing with highly sparse or dynamically evolving graphs.
- UGCN (Han et al., 2026) employs both local and global graph information to address node- and graph-level clustering collaboratively, aligning the two levels through joint contrastive objectives.
- SMGCN (Hu et al., 2026) is the first work to address the graph-level clustering task with missing relations, which reconstructs potential edges via low-rank matrix completion and forces same-cluster graphs to converge to anchors obtained by Hungarian cluster matching.

D Fusion Module

Table 9 exhibits the fusion module ablation results on graph-level clustering tasks. From the observations, we can see that PIDGC achieves the best performance compared with all variants, validating the effectiveness of the fusion module in PIDGC. This phenomenon is consistent with the performance on the node-level task.

E Distance Measures

The performance of distance measures ablation on the graph-level clustering task is shown in Table 10. From the observations, PIDGC achieves the best performance on four out of the six datasets, while attaining the second-best performance on the remaining two datasets. Meanwhile, Wasserstein and Jaccard demonstrate promising semantic mining capabilities on specific datasets.

F Sensitivity

We conduct additional sensitivity experiments on the graph-level clustering task. Fig. 4 exhibits the sensitivity performance on graph-level clustering. These observations help identify more suitable configurations of hyperparameters when deploying different training and testing datasets.

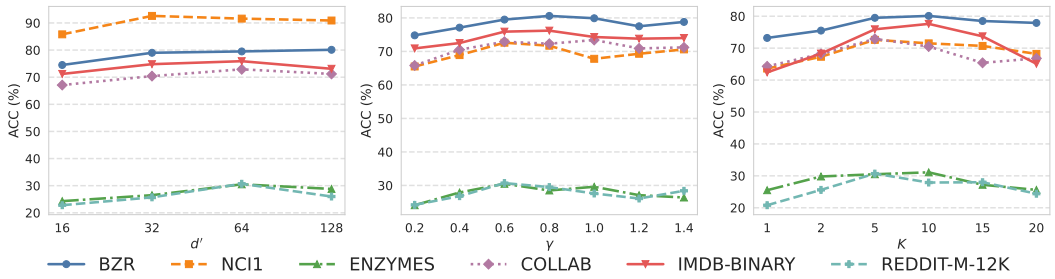


Figure 4: Sensitivity experiments on the graph-level clustering task.

Algorithm 1 Algorithm of PIDGC.

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathbf{X}, \mathbf{A})$; Epochs E.

```
Initialize the parameters of PIDGC.
for e ∈ (1, 2, ··· , E) do
    Obtain augmented features and generate the complementary graph view;
    Obtain the representations for original features and augmented features in different views;
    Calculate the loss function Eq. (7);
    Generate the integrated representations via Eq. (5);
    Calculate the clustering loss function via Eq. (9);
    Calculate the total loss function via Eq. (10);
    Update the model's parameters.
end for
return Scores
```

G PSEUDO-CODE

The algorithm is shown in Algorithm 1.

H LIMITATION

PIDGC utilizes multiple contrastive learning methods to explore multiple interactions, which results in the computational cost is relatively high. Besides, due to this limitation, we don't extend to the condition with three views. It can be seen that when the number of views exceeds 2, the computational cost increases exponentially.